Army Research Laboratory

The Maximin Adaptive-Array Algorithm for Direct-Sequence Systems

by Don Torrieri

ARL-TR-3591 August 2005

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-3591 **August 2005**

The Maximin Adaptive-Array Algorithm for Direct-Sequence Systems

Don Torrieri
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
August 2005

2. REPORT TYPE
Final

3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE
The Maximin Adaptive-Array Algorithm for Direct-Sequence Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Don Torrieri

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRD-ARL-CI-C
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-3591

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

An adaptive-array algorithm that suppresses interference in a direct-sequence system is proposed. The *maximin* algorithm differs from alternative algorithms in that it requires neither training sequences, directional information, decision-directed adaptation, nor elaborate computations such as eigenanalysis. Simulation experiments confirm that the algorithm supplements the direct-sequence processing gain with a large amount of additional interference suppression. Although the maximin algorithm is effective when code synchronization exists, strong directional interference can prevent code acquisition in a direct-sequence system, thereby rendering the system inoperable. Previously proposed adaptive-array algorithms designed to enhance code acquisition are incapable of suppressing directional interference and may even amplify it. The recursive suppression algorithm is shown to dramatically reduce the interference level relative to that of a desired direct-sequence signal before code acquisition has been achieved. As a result, the code acquisition process is greatly facilitated.

15. SUBJECT TERMS
Adaptive array, direct sequence, interference suppression, acquisition

16. SECURITY CLASSIFICATION OF:

a. REPORT
Unclassified

b. ABSTRACT
Unclassified

c. THIS PAGE
Unclassified

17. LIMITATION OF ABSTRACT
SAR

18. NUMBER OF PAGES
42

19a. NAME OF RESPONSIBLE PERSON
Don Torrieri

19b. TELEPHONE NUMBER (*Include area code*)
(301) 394-2484

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

The *maximin algorithm* is an adaptive-array algorithm that suppresses interference and thereby supplements the inherent processing gain of a spread-spectrum system. Originally developed for frequency-hopping systems *(1)*, the algorithm discriminates between the desired signal and interference on the basis of the distinct spectral characteristics of spread-spectrum signals. As indicated by its name, the maximin algorithm simultaneously maximizes the desired-signal component and minimizes the interference component in the despread signal. An adaptive array using the maximin algorithm provides a direct-sequence system with a high degree of protection against strong interference that would overwhelm the inherent processing gain. The main advantages of the maximin algorithm relative to its alternatives (e.g., *(2)*,*(3)*,*(4)*) are that it does not require training sequences, directional information, decision-directed adaptation, or elaborate computations such as eigenanalysis.

The basic configuration of an adaptive array for spread-spectrum systems is displayed in figure 1. The spreading waveform, which is produced by a synchronized receiver, enables the despreading of each copy of the spread-spectrum signal. The sample values of the complex envelopes of the despread signals are extracted by initial processors and applied to an adaptive processor that executes the maximin algorithm. In a direct-sequence receiver, the despread desired signal has a narrowband spectrum, but the interference has a wideband spectrum. This spectral difference is exploited by the maximin algorithm to estimate the interference and then cancel it.

The next section introduces the notation and provides a derivation of the maximin algorithm. In section 3, the details of the implementation of the maximin processor in a direct-sequence system are explained. The convergence analysis of section 4 establishes bounds on the adaptation constant and justifies the adaptation sequence used in the maximin algorithm. Section 5 presents the results of simulation experiments. The derivation, convergence analysis, and simulation results for the recursive suppression algorithm, which is applicable prior to code acquisition, are presented in section 6.

2. Derivation of Maximin Algorithm

The desired signal and the interference are assumed to arrive at an adaptive array of N antennas. The desired signal, interference signals, and thermal noise are modeled as independent zero-mean, wide-sense-stationary stochastic processes. Let $\mathbf{x}(i)$ denote the discrete-time vector of the complex envelopes of the N antenna outputs after each one has been sampled, despread, and filtered. The index i denotes the sample number. The vector

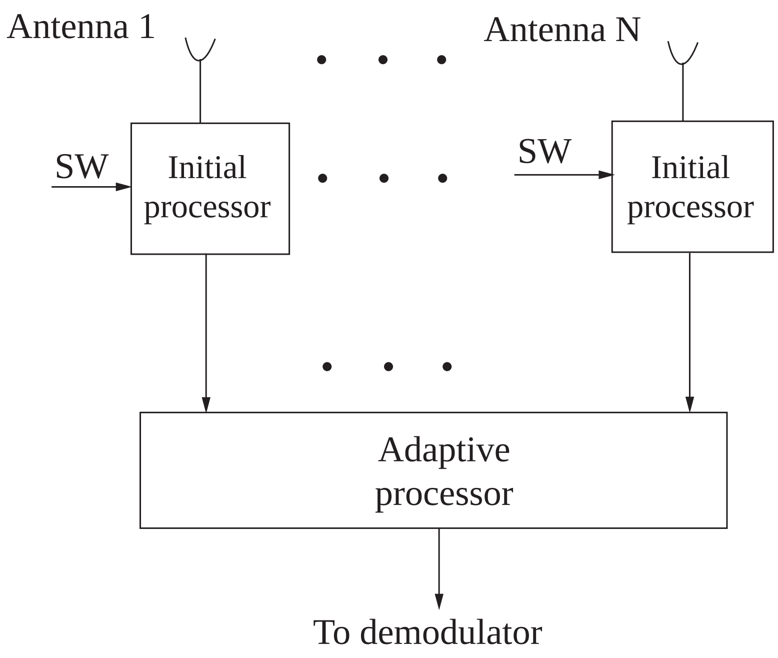

Figure 1. Architecture of adaptive array for spread-spectrum system. SW = spreading waveform.

$\mathbf{x}(i)$ can be decomposed as

$$\mathbf{x}(i) = \mathbf{s}(i) + \mathbf{n}(i) \tag{1}$$

where $\mathbf{s}(i)$ is the vector of desired-signal complex envelopes, and $\mathbf{n}(i)$ is the vector of interference and thermal-noise complex envelopes. The adaptive filter generates a weight vector $\mathbf{W}$ with complex-valued components. The output of the adaptive filter is

$$y(i) = \mathbf{W}^H \mathbf{x}(i) = y_s(i) + y_n(i) \tag{2}$$

where H denotes the complex transpose and

$$y_s(i) = \mathbf{W}^H \mathbf{s}(i), \qquad y_n(i) = \mathbf{W}^H \mathbf{n}(i). \tag{3}$$

Let $E[\ ]$ denote the expected value. When $\mathbf{W}$ is a constant, the desired-signal output power is

$$p_s = E[|y_s(i)|^2] = \mathbf{W}^H \mathbf{R}_s \mathbf{W} \tag{4}$$

where $\mathbf{R}_s$ is the desired-signal correlation matrix:

$$\mathbf{R}_s = E[\mathbf{s}(i)\mathbf{s}^H(i)]. \tag{5}$$

The interference-plus-noise output power is

$$p_n = \mathbf{W}^H \mathbf{R}_n \mathbf{W} \tag{6}$$

where $\mathbf{R}_n$ is the interference-plus-noise correlation matrix:

$$\mathbf{R}_n = E[\mathbf{n}(i)\mathbf{n}^H(i)]. \tag{7}$$

The signal-to-interference-plus-noise ratio (SINR) is

$$\rho = \frac{p_s}{p_n} = \frac{\mathbf{W}^H\mathbf{R}_s\mathbf{W}}{\mathbf{W}^H\mathbf{R}_n\mathbf{W}}\ . \tag{8}$$

The SINR provides the performance measure that the adaptive algorithm seeks to maximize.

The weight vector may be decomposed as $\mathbf{W} = \mathbf{W}_R + j\mathbf{W}_I$, where $\mathbf{W}_R$ and $\mathbf{W}_I$ are the real and imaginary parts of $\mathbf{W}$, respectively, and $j = \sqrt{-1}$. Let ∇_{WR} and ∇_{WI} denote the gradients with respect to $\mathbf{W}_R$ and $\mathbf{W}_I$, respectively. The complex gradient is defined as

$$\nabla_w = \frac{1}{2}\left(\nabla_{WR} + j\nabla_{WI}\right). \tag{9}$$

The maximin algorithm is based on the *method of steepest descent (5),(6).* In this method, the weight vector is changed along the direction of the negative gradient of a performance measure that is to be minimized. Separate steepest-descent equations can be written for $\mathbf{W}_R$ and $\mathbf{W}_I$. Combining these equations and using the negative SINR as the performance measure, we obtain

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu_0(k)\nabla_w\rho(k) \tag{10}$$

where k denotes the weight iteration number, $\mu_0(k)$ is a scalar sequence that controls the rate of change of the weight vector, and $\nabla_w\rho(k)$ is the gradient of the SINR at iteration k. Equations 8 and 9 yield

$$\nabla_w\rho = \rho\left[\frac{\mathbf{R}_s\mathbf{W}}{p_s} - \frac{\mathbf{R}_n\mathbf{W}}{p_n}\right]. \tag{11}$$

Substitution of this equation into equation 10 and the replacement of $\mathbf{W}$ with $\mathbf{W}(k)$ gives the steepest-descent algorithm. However, this algorithm requires knowledge of $\mathbf{R}_s$ and p_s, which are difficult to estimate directly. To obtain a more practical algorithm, we observe that if the interference and noise are zero-mean and statistically independent of the desired signal, then the *input correlation matrix* is

$$\mathbf{R}_x = E[\mathbf{x}(i)\mathbf{x}^H(i)] = \mathbf{R}_s + \mathbf{R}_n \tag{12}$$

and the adaptive-filter output power is

$$p_x = E[|y(i)|^2] = \mathbf{W}^H\mathbf{R}_x\mathbf{W} = p_s + p_n. \tag{13}$$

Substitution of equations 12 and 13 into 11 and simplification yields

$$\nabla_w\rho = (\rho+1)\left[\frac{\mathbf{R}_x\mathbf{W}}{p_x} - \frac{\mathbf{R}_n\mathbf{W}}{p_n}\right] \tag{14}$$

which may be approximated by observing that if $\mathbf{W}$ is modeled as deterministic (nonrandom), then $\mathbf{R}_x\mathbf{W} = E[\mathbf{x}(i)y^*(i)]$ and $\mathbf{R}_n\mathbf{W} = E[\mathbf{n}(i)y_n^*(i)]$, where the asterisk denotes the complex conjugate. Thus, we only require estimators of $E[\mathbf{x}(i)y^*(i)]$, $E[\mathbf{n}(i)y_n^*(i)]$, p_x, and p_n.

A further simplification that ultimately reduces the amount of computation by nearly a factor of two is obtained by assuming that $\mathbf{x}(i)$ is obtained by filtering samples of a continuous-time vector $\mathbf{X}(t)$ of complex envelopes. Each component of $\mathbf{X}(t)$ is a delayed version of the first component, which serves as a reference signal and may be expressed as

$$X_1(t) = X_c(t) + jX_s(t) \tag{15}$$

where $X_c(t)$ and $X_s(t)$ are the real and imaginary parts of $X_1(t)$. If $X_1(t)$ is modeled as a zero-mean, wide-sense stationary process, then the autocorrelations of $X_c(t)$ and $X_s(t)$ are identical, and the circular symmetry implies that $E[X_c(t)X_s(t+\tau)] = -E[X_s(t)X_c(t+\tau)]$, where τ is an arbitrary delay *(7)*, *(8)*. From these relations, it follows that

$$E[\mathbf{x}(i)\mathbf{x}^T(i)] = \mathbf{0} \tag{16}$$

where T denotes the transpose. The adaptive-filter output can be decomposed as

$$y(i) = y_r(i) + jy_i(i) \tag{17}$$

where $y_r(i)$ and $y_i(i)$ are the real and imaginary parts of $y(i)$, respectively. If the weight vector is modeled as deterministic, then equations 2 and 16 imply that

$$E[\mathbf{x}(i)y_r(i)] = E\left[\mathbf{x}(i)\left\{\frac{1}{2}\mathbf{x}^H(i)\mathbf{W} + \frac{1}{2}\mathbf{x}^T(i)\mathbf{W}^*\right\}\right] = \frac{1}{2}E\left[\mathbf{x}(i)\mathbf{x}^H(i)\right]\mathbf{W}. \tag{18}$$

This equation and equation 12 yield

$$\mathbf{R}_x\mathbf{W} = 2E[\mathbf{x}(i)y_r(i)]. \tag{19}$$

Similarly, if zero-mean, wide-sense stationary interference arriving at each antenna is a delayed version of the interference arriving at antenna 1, and if the zero-mean thermal noise is independent in each array branch, then

$$E[\mathbf{n}(i)\mathbf{n}^T(i)] = \mathbf{0} \tag{20}$$

and, hence,

$$\mathbf{R}_n\mathbf{W} = 2E[\mathbf{n}(i)y_{nr}(i)] \tag{21}$$

where $y_{nr}(i)$ is the real part of $y_n(i)$. Straightforward calculations using equations 2, 16, and 13 yield $E[y_r^2(i)] = E[y_i^2(i)]$ and $E[y_r(i)y_s(i)] = 0$. Therefore, equations 13 and 17 imply that $p_x = 2E[y_r^2(i)]$. Similarly, $p_n = 2E[y_{nr}^2(i)]$. Substitution of the preceding results into sample values of equation 14 yields

$$\nabla_w \rho = (\rho + 1) \left\{ \frac{E[\mathbf{x}y_r]}{E[y_r^2]} - \frac{E[\mathbf{n}y_{nr}]}{E[y_{nr}^2]} \right\} \tag{22}$$

where the sample index has been omitted for simplicity.

To derive the maximin algorithm, let $\hat{p}_x(k)$ and $\hat{p}_n(k)$ denote estimates of $E[y_r^2]$ and $E[y_{nr}^2]$, respectively, following weight iteration k. The estimate of ρ following iteration k is $\widehat{\rho}(k)$. Let $\mathbf{C}_x(k)$ and $\mathbf{C}_n(k)$ denote estimates following iteration k of the input correlation vector $E[\mathbf{x}y_r]$ and the noise correlation vector $E[\mathbf{n}y_{nr}]$, respectively. The *adaptation sequence* is defined as $\alpha(k) = \mu_0(k)[\widehat{\rho}(k) + 1]$. Substituting these estimates into equations 22 and 10, we obtain the basic form of the maximin algorithm:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \alpha(k) \left[\frac{\mathbf{C}_x(k)}{\hat{p}_x(k)} - \frac{\mathbf{C}_n(k)}{\hat{p}_n(k)} \right], \qquad k \geq 0 \tag{23}$$

where $\mathbf{W}(0)$ is the deterministic initial weight vector. As the adaptive weights converge, the interference components of $\mathbf{C}_x(k)$ and $\hat{p}_x(k)$ decrease. Thus, the first term within the brackets can be interpreted as a *signal term* that enables the algorithm to direct the array beam toward the desired signal. The second term within the brackets is a *noise term* that enables the algorithm to null interference signals.

The adaptation sequence $\alpha(k)$ should be chosen so that $E[\mathbf{W}(k)]$ converges to a nearly optimal steady-state value. It is also intuitively plausible that $\alpha(k)$ should decrease rapidly as $E[\mathbf{W}(k)]$ converges. A suitable candidate is

$$\alpha(k) = \alpha \frac{\hat{p}_n(k)}{\hat{t}(k)} \tag{24}$$

where $\hat{t}(k)$ is an estimate of the total interference and noise power entering the array, and α is the *adaptation constant.* The subsequent convergence analysis and simulation results confirm that this choice is effective and robust, provided that the adaptation constant is within certain numerical bounds. Another adaptation sequence that works well *(1)* is $\alpha(k) = \alpha\hat{p}_n(k)/\hat{p}_x(k)$, but the best choice for α in this sequence depends on the SINR at each input to the adaptive filter.

The remaining issue is the choice of estimators for $\hat{t}(k), \mathbf{C}_x(k), \mathbf{C}_n(k), \hat{p}_x(k)$, and $\hat{p}_n(k)$. The specific nature of the spread-spectrum signals allows blind estimates to be made without depending on known steering vectors or reference signals.

3. Implementation of Processor

The maximin processor is placed after the despreading subsystem of the receiver. For the phase-shift keying (PSK) modulation, the principal components of the initial processor in each branch following an antenna, which includes the despreading subsystem, are shown in figure 2. The front-end devices include a bandpass filter that blocks noise outside the band occupied by the spread-spectrum signal. Assuming frequency and chip synchronization in the receiver, the received signal is downconverted and then applied to a chip matched filter. If the sampling rate equals the chip rate, the desired component of the sampled output of the chip matched filter is the data-modulated spreading sequence. If code synchronization of the spreading sequence has been established in the receiver, the final mixing operation produces the branch output sequence comprising the despread desired sequence, spectrally spread interference, and noise.

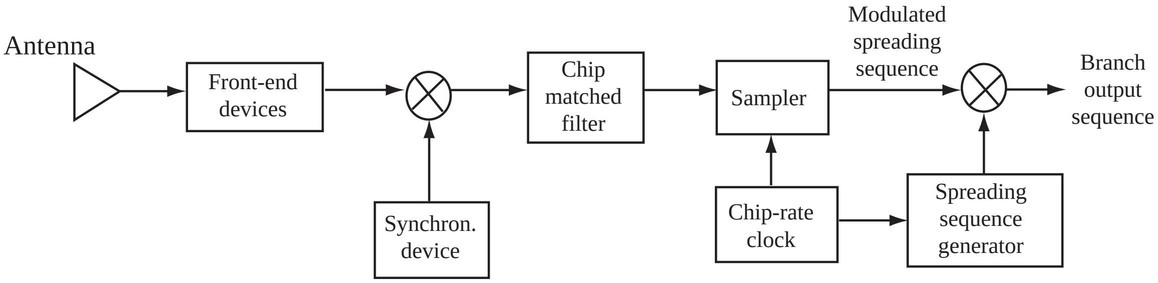

Figure 2. Initial processor in each branch of the receiver for a direct-sequence signal with PSK.

Although frequency synchronization is necessary for downconversion of the received signal to baseband, no phase synchronization is attempted in the array branches because relative phase information must be preserved for successful beamforming and interference cancellation. Phase synchronization is ultimately necessary for coherent demodulation of PSK. The random phase in the adaptive-filter output is subsequently removed by using the output of a carrier synchronization system.

In the maximin processor shown in figure 3, each branch output sequence is applied to a baseband filter (BF),which produces a component of $\mathbf{x}(i)$, and a monitor filter (MF), which estimates the interference. For an adaptive array of N antennas, the outputs of N pairs of baseband and monitor filters are applied to the adaptive filter, the output of which is applied to a digital demodulator. The maximin algorithm seeks to maximize the SINR at the input to the demodulator.

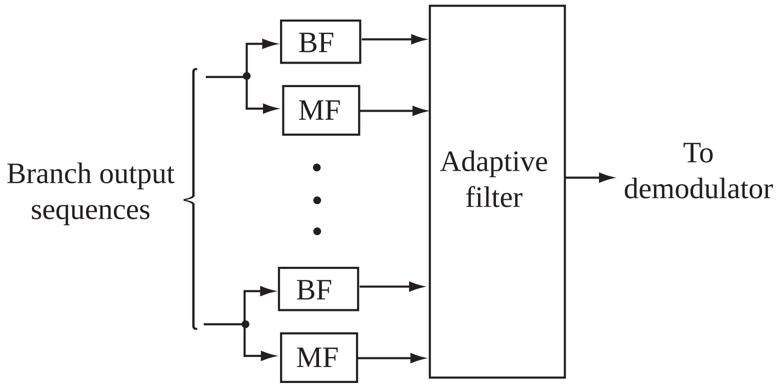

Figure 3. Maximin processor.

Figure 4 sketches the mainlobes of the frequency responses of the baseband and monitor filters. Each baseband filter has a frequency response $H(f)$ with one-sided bandwidth $B \approx 1/T_s$, where T_s is the data-symbol duration. The N baseband-filter outputs generate $\mathbf{x}(i)$. Each monitor filter has a frequency response

$$H_1(f) = \frac{1}{2}H(f - f_0) + \frac{1}{2}H(f + f_0) \tag{25}$$

where the center-frequency offset is f_0, as indicated in figure 4. The factor 1/2 ensures an accurate interference-power estimate when the interference is approximately spectrally uniform over the band $|f| \leq f_0 + B$.

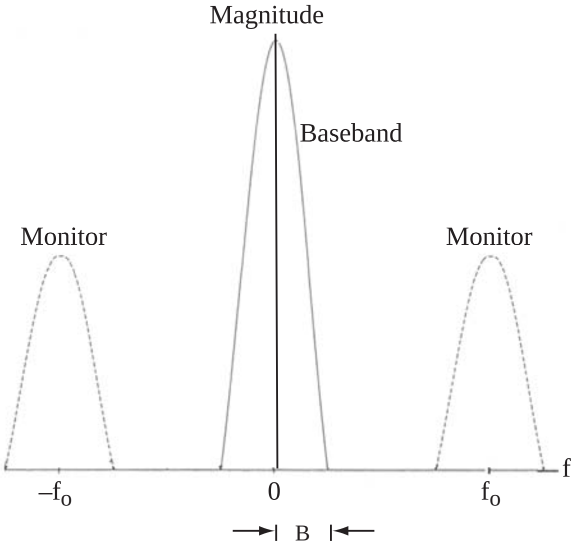

Figure 4. Frequency responses of baseband and monitor filters.

Each baseband filter can have the same transfer function it would have if only a single antenna were used. Thus, since the despread desired signal is constant during a symbol, this signal is applied to a matched filter with a rectangular impulse response. The z-domain transfer function of the matched filter is

$$H(z) = 1 + z^{-1} + \cdots + z^{-(g-1)} = \frac{1 - z^{-g}}{1 - z^{-1}} \tag{26}$$

where $g = T_s/T_c$ denotes the number of chips per symbol or the processing gain, and T_c is the chip duration. This filter has a null-to-null bandwidth equal to $2/gT_c = 2/T_s$. Each matched-filter output is sampled at the end of every symbol interval. Let $\mathbf{x}_1(\ell)$ denote the vector of branch output sequences applied to the N matched filters. The vector output of these filters is

$$\mathbf{x}(i) = \sum_{\ell=i-g+1}^{i} \mathbf{x}_1(\ell) \tag{27}$$

where ℓ is the index of the chip-rate input samples and i is the index of the symbol-rate output samples. A device that computes this sum is called an *accumulator*. Similarly, the monitor filters, each of which has frequency response $H_1(f)$, are *bandpass accumulators* that produce

$$\hat{\mathbf{n}}(i) = \sum_{\ell=i-g+1}^{i} \mathbf{x}_1(\ell)\cos(2\pi f_0 T_c \ell) \tag{28}$$

which is the vector of interference-plus-noise estimates used to generate $\mathbf{C}_n(k)$ and $\hat{p}_n(k)$.

The despreading of the direct-sequence signal spreads the spectral density of the interference over the entire passband of the monitor filter if $f_0 \leq (g-1)/T_s$. Any spillover or spectral splatter of the desired-signal spectrum into the monitor filter may lead to some degree of desired-signal cancellation by the adaptive algorithm. Thus, f_0 must be sufficiently large to prevent significant spectral splatter.

The architecture of the adaptive filter is illustrated in figure 5. The vectors applied to the adaptive filter are $\mathbf{x}(i)$ and $\hat{\mathbf{n}}(i)$, and m symbol-rate samples are taken per weight iteration. The adaptive filter produces the output

$$y_r(i) = Re[\mathbf{W}^H(k)\mathbf{x}(i)], \quad i = km+1, \cdots, (k+1)m \tag{29}$$

where sample i is taken after weight iteration k. This output is applied to the demodulator and is used in the estimators

$$\mathbf{C}_x(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m} \mathbf{x}(i) y_r(i), \qquad k \geq 0 \tag{30}$$

and

$$\hat{p}_x(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m} y_r^2(i), \qquad k \geq 0. \tag{31}$$

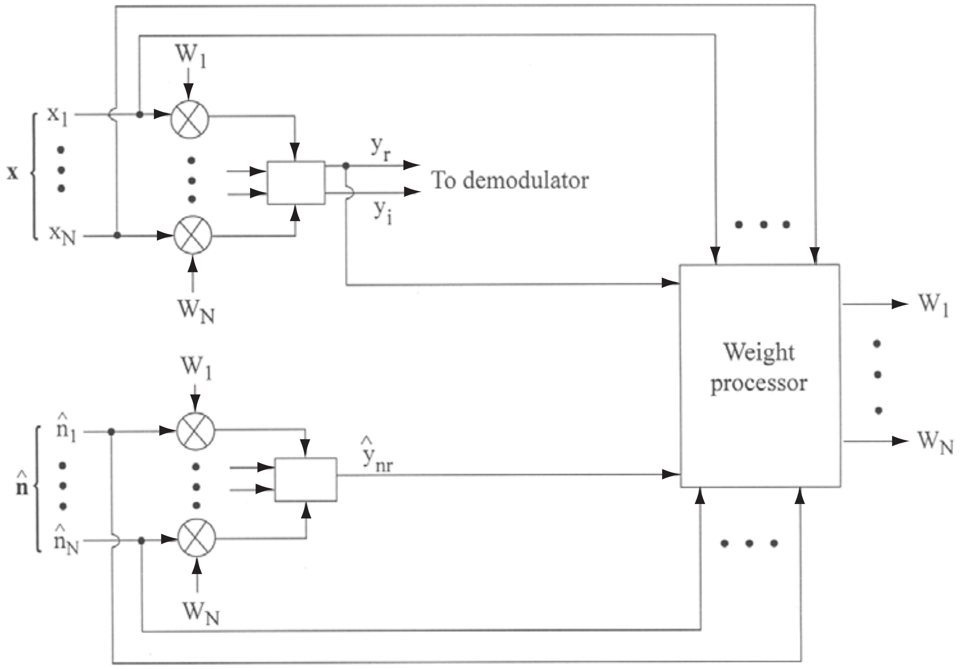

Figure 5. Adaptive filter that executes maximin algorithm.

The adaptive filter also generates

$$\hat{y}_{nr}(i) = Re[\mathbf{W}^H(k)\hat{\mathbf{n}}(i)], \quad i = km+1, \cdots, (k+1)m \; . \tag{32}$$

If $f_0 \leq (g-1)/T_s$ so that the interference power is spread approximately uniformly over $|f| \leq f_0 + B$, then the form of $\mathrm{H}_1(f)$ indicates that the interference powers at the baseband-filter and monitor-filter outputs are approximately equal. Since the noise components are zero-mean and independent and the each interference component is a phase-shifted version of the component in a reference branch, $E[\hat{\mathbf{n}}(i)\hat{\mathbf{n}}^H(i)] \approx E[\mathbf{n}(i)\mathbf{n}^H(i)]$. Therefore,

$$E[\hat{\mathbf{n}}(i)\hat{y}_{nr}(i)] \approx E[\mathbf{n}(i)y_{nr}(i)] \tag{33}$$

and a suitable estimator of the interference-plus-noise correlation at weight iteration k is

$$\mathbf{C}_n(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m} \hat{\mathbf{n}}(i)\hat{y}_{nr}(i), \qquad k \geq 0. \tag{34}$$

Similarly, a suitable estimator proportional to the interference-plus-noise output power is

$$\hat{p}_n(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m} \hat{y}^2_{nr}(i), \qquad k \geq 0. \tag{35}$$

An estimator of the total interference-plus-noise power entering the array is

$$\hat{t}(k) = \frac{1}{m}\sum_{i=km+1}^{(k+1)m} \| \hat{\mathbf{n}}(i) \|^2, \qquad k \geq 0. \tag{36}$$

where $\| \mathbf{v} \|$ denotes the Euclidean norm of $\mathbf{v}$. These estimators complete the specification of the maximin algorithm given by equations 23 and 24.

As verified by simulation experiments, recursive versions of the preceding estimators merely slow the convergence of the maximin algorithm, even if the interference statistics are stationary. Simulation experiments confirm that for cyclostationary spread-spectrum signals and tone interference, the algorithm simplification stemming from the assumption of wide-sense stationary processes causes no overall performance loss other than a slight slowing of convergence.

In a multipath environment, separate pairs of initial and maximin processors can establish a distinct adaptive-array pattern for each resolved path. A rake combiner can then maximize the SINR of the combined signal derived from all these patterns. If the array antennas are sufficiently close, the fading across the array is completely correlated, and the adaptive processing may precede the channel estimation required by the rake combiner.

4. Convergence Analysis

Let $s(i)$ denote the component of $\mathbf{s}(i)$ derived from a fixed reference antenna. In practical spread-spectrum systems, the desired signal is sufficiently narrowband that its copies in all the branches are nearly aligned in time. Therefore,

$$\mathbf{s}(i) = s(i)\mathbf{S}_0 \tag{37}$$

where $\mathbf{S}_0$ is a *steering vector* of complex numbers that represent the relative amplitudes and phase shifts at the antenna outputs. The substitution of this equation into equation 5 yields

$$\mathbf{R}_s = p_{si}\mathbf{S}_0\mathbf{S}_0^H \tag{38}$$

where

$$p_{si} = E[|s(i)|^2]. \tag{39}$$

After the substitution of these equations into equation 8, the maximization of the SINR (appendix) yields the optimal weight vector

$$\mathbf{W}_0 = \eta\mathbf{R}_n^{-1}\mathbf{S}_0 \tag{40}$$

where η is an arbitrary constant. The maximum SINR, which occurs if $\mathbf{W} = \mathbf{W}_0$, is

$$\rho_0 = p_{si}\mathbf{S}_0^H\mathbf{R}_n^{-1}\mathbf{S}_0. \tag{41}$$

The highly nonlinear nature of the maximin algorithm precludes a completely rigorous convergence analysis. However, with enough approximations, the convergence of the mean weight vector to $\mathbf{W}_0$ can be demonstrated and bounds on the adaptation constant can be derived. We assume that the interference is wide-sense stationary and m is large enough that equation 36 gives

$$\hat{t}(k) \approx r = E[\hat{t}(k)] = E[\parallel \hat{\mathbf{n}}(i) \parallel^2] \approx E[\parallel \mathbf{n}(i) \parallel^2] = tr(\mathbf{R}_n) \tag{42}$$

where $tr(\;)$ denotes the trace. We assume that after a number of algorithm iterations k_0, $\hat{p}_x(k)/\hat{p}_n(k) \approx \rho + 1 \approx \rho_0 + 1$. Using these assumptions in equations 23 and 24, the maximin algorithm is approximated by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\alpha}{r}\left[\frac{\mathbf{C}_x(k)}{(\rho_0+1)} - \mathbf{C}_n(k)\right], \qquad k \geq k_0. \tag{43}$$

Since $\mathbf{W}(k)$ does not depend on $\mathbf{x}(i)$ and $\widehat{\mathbf{n}}(i)$ for $i \geq km+1$, we make the approximation that $\mathbf{W}(k)$ is statistically independent of $\mathbf{x}(i)$ and $\mathbf{n}(i)$ for $i \geq km+1$. We obtain from equations 30, 29, 16, and 2 that

$$E[\mathbf{C}_x(k)] = E[\mathbf{x}(i)y_r(i)] = \frac{1}{2}\mathbf{R}_x E[\mathbf{W}(k)]. \tag{44}$$

Similarly, equations 34, 33, 32, 20, and 3 yield

$$E[\mathbf{C}_n(k)] = E[\hat{\mathbf{n}}(i)\hat{y}_{nr}(i)] \approx \frac{1}{2}\mathbf{R}_n E[\mathbf{W}(k)]. \tag{45}$$

Taking the expected value of both sides of equation 43, substituting equations 45, 44, and 12, and simplifying algebraically, we obtain the approximate recursive equation for the mean weight vector:

$$E[\mathbf{W}(k+1)] = \left[\mathbf{I} - \frac{\alpha}{2r(\rho_0+1)}\mathbf{D}\right] E[\mathbf{W}(k)], \qquad k \geq k_0 \tag{46}$$

where

$$\mathbf{D} = \rho_0\mathbf{R}_n - \mathbf{R}_s = \rho_0\mathbf{R}_n - p_{si}\mathbf{S}_0\mathbf{S}_0^H. \tag{47}$$

A straightforward calculation using equations 47 and 41 yields

$$\mathbf{D}\;\mathbf{R}_n^{-1}\mathbf{S}_0 = \mathbf{0} \tag{48}$$

which indicates that $\mathbf{R}_n^{-1}\mathbf{S}_0$ is an eigenvector of $\mathbf{D}$, and the corresponding eigenvalue is 0. Since $\mathbf{D}$ is Hermitian, it has a complete set of N orthogonal eigenvectors, one of which is $\mathbf{R}_n^{-1}\mathbf{S}_0$. Since $\rho \leq \rho_0$, equation 8 implies that $\mathbf{W}^H\mathbf{R}_s\mathbf{W} \leq \rho_0\mathbf{W}^H\mathbf{R}_n\mathbf{W}$. Consequently, $\mathbf{W}^H\mathbf{D}\ \mathbf{W} \geq 0$, which proves that $\mathbf{D}$ is positive semidefinite and, hence, has N nonnegative eigenvalues. Since only $\mathbf{W} = \mathbf{W}_0$ gives $\rho = \rho_0$, one of these eigenvalues is zero, and the other $N-1$ eigenvalues are positive.

To solve equation 46, we make the decomposition

$$E[\mathbf{W}(k)] = \eta(k)\mathbf{R}_n^{-1}\mathbf{S}_0 + \sum_{i=2}^{N} a_i(k)\mathbf{e}_i \tag{49}$$

where each $a_i(k)$ and $\eta(k)$ are scalar functions and $\mathbf{e}_i$ is one of the $N-1$ eigenvectors orthogonal to $\mathbf{R}_n^{-1}\mathbf{S}_0$. Substituting this equation into equation 46 and using the orthogonality of the eigenvectors, we obtain

$$\eta(k+1) = \eta(k) = \eta(k_0), \qquad k \geq k_0 \tag{50}$$

$$a_i(k+1) = \left[1 - \frac{\alpha\lambda_i}{2r(\rho_0+1)}\right] a_i(k), \quad 2 \leq i \leq N, \qquad k \geq k_0 \tag{51}$$

where λ_i is the eigenvalue corresponding to $\mathbf{e}_i$. Assuming that $\eta(k_0) \neq 0$, equations 40, 49, and 50 indicate that $E[\mathbf{W}(k)] \longrightarrow \mathbf{W}_0$ as $k \to \infty$ if and only if each $a_i(k) \to 0$. The solution to equation 51 is

$$a_i(k) = \left[1 - \frac{\alpha\lambda_i}{2r(\rho_0+1)}\right]^{k-k_0} a_i(k_0), \quad 2 \leq i \leq N, \qquad k \geq k_0. \tag{52}$$

This equation indicates that $a_i(k) \longrightarrow 0$, $2 \leq i \leq N$, as $k \to \infty$ if and only if

$$\left|1 - \frac{\alpha\lambda_i}{2r(\rho_0+1)}\right| < 1\ , \quad 2 \leq i \leq N. \tag{53}$$

This inequality implies that the necessary and sufficient condition for the convergence of the mean weight vector is

$$0 < \alpha < \frac{4r(\rho_0+1)}{\lambda_{max}} \tag{54}$$

where λ_{max} is the largest eigenvalue of $\mathbf{D}$.

Since the sum of the eigenvalues of a square matrix is equal to its trace,

$$\lambda_{max} \leq \sum_{i=1}^{N} \lambda_i = tr(\mathbf{D}) = \rho_0 r - tr(\mathbf{R}_s) \leq \rho_0 r. \tag{55}$$

Substituting this bound into equation 54 and simplifying the result, we obtain

$$0 < \alpha < 4 \tag{56}$$

as a sufficient (but not necessary) condition for the convergence of the mean weight vector to the optimal weight vector. Although this inequality must be regarded as an approximation because of the approximations used in its derivation, it gives at least rough guidance in the selection of the adaptation constant. The fact that the upper bound is numerical and does not depend on environmental parameters provides support for the choice of equation 24 as the adaptation sequence.

5. Simulation of Maximin Algorithm

In the simulation experiments, the array consists of four omnidirectional antennas located at the vertices of a square or in a uniform linear configuration. Let λ denote the wavelength corresponding to the center frequency of the desired signal, which is 3 GHz. The edge length or the separation between adjacent antennas is $d = 0.5\lambda$, $d = 1.0\lambda$, or $d = 1.5\lambda$. The direct-sequence signal with PSK modulation and a rectangular chip waveform arrives from a direction 20 degrees clockwise from the perpendicular to one of the edges. All signals are assumed to arrive as plane waves with no fading. The direct-sequence signal has a frequency offset equal to 1 kHz after downconversion, which models imperfect frequency synchronization. The data-bit and spreading sequences are randomly generated for each simulation trial at the rates of 100 kbps and 10 Mbps, respectively. The processing gain is 20 dB. Perfect chip and spreading-sequence synchronization in the receiver are assumed. As indicated in figure 2, the sampling is done once per spreading-sequence chip. The thermal noise in each branch output is modeled as bandlimited white Gaussian noise. The signal-to-noise ratio (SNR) is 0 dB at the input of each of the baseband and monitor filters. The baseband filters are accumulators. The monitor filters are bandpass accumulators offset by $f_0 = 400$ kHz to prevent contamination by the direct-sequence signal. The maximin algorithm is implemented with $\alpha = 1$. A weight iteration occurs after each ten data bits. Each simulation trial represents the execution of the algorithm until it terminates. For each trial, the initial weight vector of the adaptive processor is $\mathbf{W}(0) = [1\ \ 0\ \ 0\ \ 0]$, which forms an omnidirectional array pattern. The principal system parameter values are listed in table 1.

Each of 1, 2, or 3 interference signals is a tone (continuous-wave signal). After the downconversions, the tones have different initial phase shifts and residual frequency offsets equal to 10 kHz, which reflects the mismatch of the tone frequencies and the carrier frequency of the direct-sequence signal. Multiple tones do not add coherently, even if they have the same carrier frequencies, because they arrive from different directions and have different initial phase shifts. The SINR at the processor output is calculated after each sample time and then averaged over all samples in the time interval between a weight iteration and a preceding weight iteration to determine the SINR at each weight iteration.

Table 1. Basic system parameters.

Parameter	**Value**
Array antennas	4
Array geometry	square or linear
Center frequency	3 GHz
Antenna separation	0.5λ, 1.0λ, or 1.5λ
Direction of desired signal	20°
SNR in each branch output	0 dB
Modulation	PSK
Data rate	100 kbs
Filter bandwidths	100 kHz
Monitor filter offset	400 kHz
Frequency offset of desired signal	1 kHz
Chip and sampling rate	10 Mbs
Bits per weight iteration	10
Weight iterations per trial	100
Adaptation constant	$\alpha = 1$
Initial weight vector	$[1 \quad 0 \quad 0 \quad 0]$
Frequency offset of interference	10 kHz

The SINR is observed to fluctuate, but tends to gradually increase until it reaches a steady-state condition with a smaller residual fluctuation.

Figure 6 illustrates the SINR variation versus the weight iteration number for a typical simulation trial in which one interference tone arrives at a 50° angle with an interference-to-signal ratio (ISR) equal to 30 dB. Let θ denote an arrival angle defined as the angle in the counterclockwise direction from the normal to one of the array edges. Let $\mathbf{S}(\theta)$ denote the *array response vector*, which is the array response to an ideal plane wave arriving at angle θ *(5)*, *(8)*. The array gain pattern after weight iteration k is

$$G(\theta, k) = \left|\mathbf{W}^H(k)\mathbf{S}(\theta)\right|^2 . \tag{57}$$

The array gain pattern at the end of the simulation trial of figure 6 is depicted in figure 7. A null deeper than −20 dB in the direction of the interference signal and a mainlobe slightly displaced from the direction of the desired signal have formed along with other grating nulls and grating lobes.

The results of 15 representative simulation experiments using the parameter values of table 1 are summarized in table 2. Each experiment comprises 50 trials with 100 weight iterations per trial. The first column gives the arrival angles of 1, 2, or 3 interference signals relative to the desired-signal direction. The ISR for each interference signal is 10 dB. The second column gives the array type: square with $d = 1.0\lambda$, square with $d = 1.5\lambda$, or linear with $d = 0.5\lambda$. The SINRs expressed in decibels for the last 20 weight iterations of all the trials are used to compute the *steady-state SINR* and the *standard*

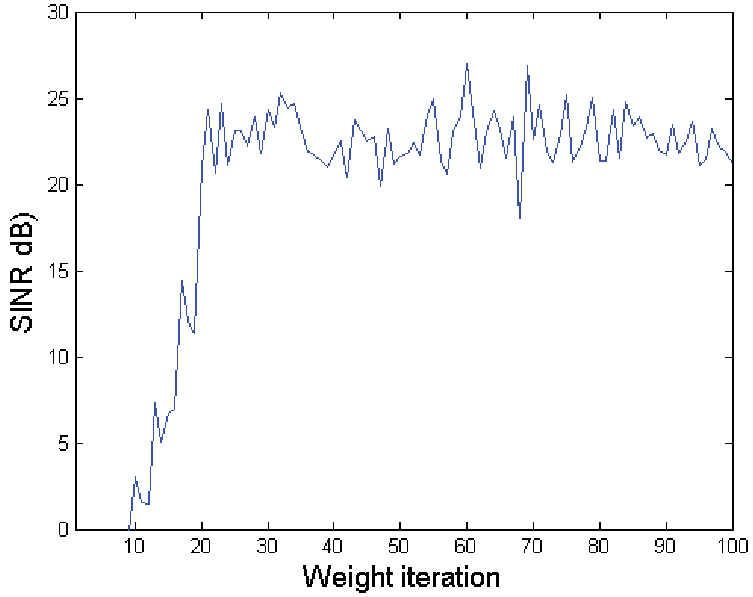

Figure 6. SINR variation in typical simulation trial for maximin algorithm, one interference tone at 50°, and ISR = 30 dB.

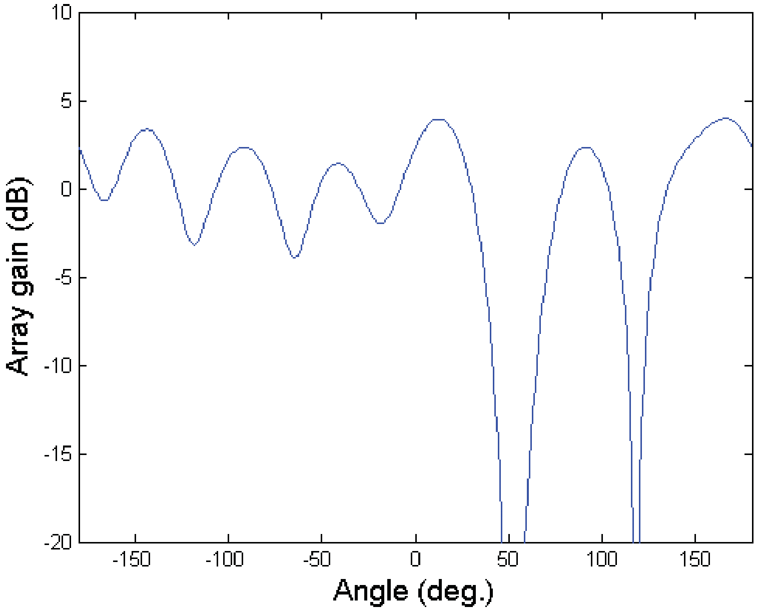

Figure 7. Array gain pattern at end of typical simulation trial for maximin algorithm, one interference tone at 50°, and ISR = 30 dB.

Table 2. Simulation results for interference tones, each with ISR = 10 dB.

Arrival angles of interf. (°)	**Array Type**	**Steady-state SINR (dB)**	**Standard Dev. (dB)**	**Crossing Number**
60	Sq, 1.0λ	25.81	1.49	3.2
60, −40	Sq, 1.0λ	22.84	2.22	8.9
60, 85	Sq, 1.0λ	24.53	1.60	5.0
60, −40, 85	Sq, 1.0λ	24.39	1.71	5.5
30	Sq, 1.0λ	20.10	1.57	16.1
60	Sq, 1.5λ	25.13	1.46	4.2
60, −40	Sq, 1.5λ	25.38	1.50	3.9
60, 85	Sq, 1.5λ	24.40	1.47	5.4
60, −40, 85	Sq, 1.5λ	24.26	1.52	5.4
30	Sq, 1.5λ	22.95	1.99	7.1
60	Linear	25.75	1.43	3.3
60, −40	Linear	24.98	1.46	4.1
60, 85	Linear	24.81	1.52	4.4
60, −40, 85	Linear	24.67	1.59	4.8
30	Linear	20.25	1.55	10.1

deviation of the SINR. The final column gives the *crossing number*, which is the average number of weight iterations required for the SINR to exceed 20 dB. The crossing number provides a rough measure of the relative time required for convergence to the steady state.

Table 2 indicates that beamforming and interference cancellation are achieved in a wide variety of scenarios. Additional interference signals do not necessarily slow the algorithm convergence or lower the steady-state SINR. The reason is that as the array forms one pattern lobe or null, it tends to form additional grating lobes and nulls and thereby may counteract the maximin algorithm. The results for an interference signal with a 30° arrival angle and a square array illustrate the limitations imposed by the resolution of the array. The *resolution*, which is the angular separation between the interference and desired-signal sources that can be accommodated without a large performance degradation, decreases as the array aperture increases. The square array with $d = 1.0\lambda$ and the uniform linear array with $d = 0.5\lambda$, despite having an array aperture equal to 1.5λ, have insufficient resolution for this interference. Increasing the antenna separation in a square array to 1.5λ provides a large improvement in the convergence speed and the steady-state SINR.

Table 3 lists the simulation results for a square array with $d = 1.0\lambda$, one interference tone with a 60° arrival angle, and a variable ISR. Each experiment comprises 50 trials with 100 weight iterations per trial. The parameters of table 1 are used except that there are 15 bits per weight iteration when ISR = 35 dB. The enormous interference-cancellation capability of the maximin processor, which supplements the inherent processing gain of the direct-sequence signal, is evident. As the ISR increases from 10 dB to 35 dB, the

Table 3. Simulation results for a square array with $d = 1.0\lambda$ and one interference tone with a 60° arrival angle.

ISR (dB)	**Steady-state SINR (dB)**	**Standard dev. (dB)**	**Crossing number**
10	25.81	1.49	3.2
20	24.22	1.44	6.2
30	23.64	1.44	18.2
35	23.66	2.61	44.3

convergence slows. The delay corresponding to the crossing number is 1.82 ms when ISR = 30 dB, but partially because of the decrease in the weight iteration rate, this delay increases to 6.645 ms when ISR = 35 dB.

As the ISR increases beyond 30 dB, it is found that the convergence of the SINR to the steady state for all trials requires an increase in the number of bits per weight iteration. In general, an increase in the number of bits per weight iteration enhances stable convergence and reduces the standard deviation of the SINR. However, the steady-state SINR is reduced, and the convergence is slower. The choice of 10 bits per weight iteration in table 1 represents a reasonable compromise in the performance objectives for ISRs that do not exceed 30 dB.

6. Acquisition

A direct-sequence signal cannot be despread until the receiver can establish a local spreading sequence synchronized with the received spreading sequence *(8)*. Prior to initial sequence synchronization or *code acquisition*, the usual adaptive-array algorithm for interference suppression, such as the maximin algorithm, might be ineffective against strong interference because the relative strengthening of the desired signal by means of despreading is unavailable. The wide variety of techniques that require only a single antenna output to reject narrowband interference *(8)* rely on sequence synchronization and, thus, are not useful prior to code acquisition. Adaptive beamforming potentially would increase the desired-signal power relative to the interference, but the direction-of-arrrival estimation and the avoidance or rejection of beams in the directions of strong interference signals might be excessively time consuming.

Typical receivers for direct-sequence signals only activate a single antenna during code acquisition *(2)*. A proposed algorithm to enhance acquisition by using the full array *(9)* is extremely computationally intensive and requires short spreading sequences, which are usually unacceptable. A more recent proposed algorithm *(10)* that uses the normalized least-mean-square algorithm does not have these disadvantages, but neither this algorithm nor the other one can prevent beamforming in the direction of strong interference. Thus, the algorithms may amplify the interference instead of cancelling it. Another approach is to use the *recursive suppression algorithm* prior to code acquisition for the purpose of interference suppression without significant desired-signal cancellation. The reduced interference in the adaptive-processor output, which is applied to the acquisition system, facilitates the code acquisition process.

The recursive suppression algorithm is designed to cancel all received signals. If the interference signals are stronger than the desired signal, they are cancelled first, which enables the acquisition system to achieve code synchronization. The algorithm terminates before it has enough time to cancel the desired signal. If acquisition has been verified, the receiver activates both the tracking system and the maximin algorithm. If acquisition has not been verified, the recursive suppression algorithm is repeated periodically with an initial omnidirectional weight vector. If the desired signal is stronger than the interference, then the algorithm is not useful, but the acquisition system only needs to process the output of a single antenna, which it does in the interims between algorithm repetitions.

Each antenna output is applied to a separate initial processor that produces a branch output sequence, as illustrated in figure 2 for a direct-sequence signal with PSK modulation. The first mixer downconverts the received bandpass signal to a complex-valued baseband signal. After the downconversion, a chip matched filter and sampler produce the data-modulated spreading sequence. This sequence is mixed with the local spreading sequence to produce a chip-rate branch output sequence that becomes the despread sequence after code acquisition has occurred.

As illustrated in figure 8 for complex-valued signals, branch output sequences are applied directly to an adaptive filter that executes the recursive suppression algorithm and is embedded within a serial-search or digital matched-filter acquisition correlator *(8)*. The adaptive filter applies the weights computed by the recursive suppression algorithm to the branch output sequences. The complex-valued adaptive-filter output is applied to an accumulator in a serial-search system or digital transversal filters in a matched-filter system. The acquisition correlator produces a decision statistic that is compared to a threshold. Because of the mixing with the local spreading sequence, both the noise and the interference at the input to the accumulator or transversal filters approximate white Gaussian processes, and the analytical results of *(8)* for calculating the acquisition performance statistics are applicable.

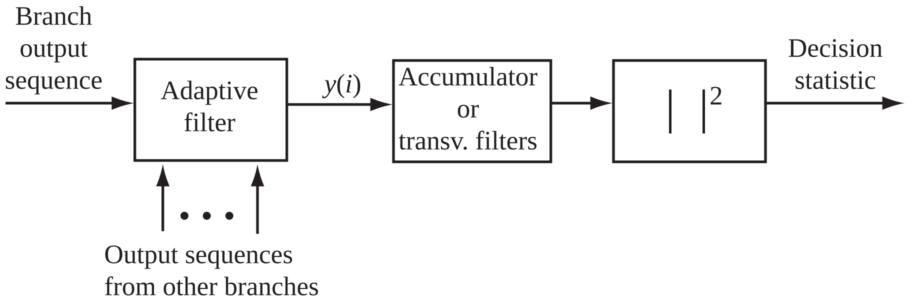

Figure 8. Adaptive filter and acquisition correlator.

6.1 Derivation of Recursive Suppression Algorithm

The recursive suppression algorithm can be derived by the method of steepest descent with the constrained minimization of the output power of an adaptive filter as the performance criterion. Let $\mathbf{x}(i)$ denote the discrete-time input vector, each component of which is a branch output sequence. If $\mathbf{x}(i)$ is modeled as a wide-sense stationary process and the weight vector $\mathbf{W}$ produced by the adaptive filter is modeled as deterministic, then the adaptive-filter output is

$$y(i) = \mathbf{W}^H \mathbf{x}(i) \tag{58}$$

and the mean output power is

$$P_x = \mathbf{W}^H \mathbf{R}_x \mathbf{W} \tag{59}$$

where the input correlation matrix is

$$\mathbf{R}_x = E[\mathbf{x}(i)\mathbf{x}^H(i)]. \tag{60}$$

The constraint that the norm of the weight vector must remain constant ensures that the optimal weight vector is not the zero vector. For mathematical tractability, the performance criterion is prescribed as the minimization of the mean output power P_x subject to the quadratic constraint

$$\| \mathbf{W} \|^2 = \mathbf{W}^H \mathbf{W} = 1. \tag{61}$$

To derive the optimal weight vector by the method of Lagrange multipliers, we minimize the real scalar

$$H = P_x - \gamma(\| \mathbf{W} \|^2 - 1) \tag{62}$$

where γ is a scalar Lagrange multiplier. Substituting equation 59, we find that the complex gradient of H is

$$\nabla_w H = 2\mathbf{R}_x \mathbf{W} - 2\gamma \mathbf{W}. \tag{63}$$

A necessary condition for the minimum is determined by setting this equation equal to the zero vector, which yields the following equation that must be satisfied by the optimal weight vector $\mathbf{W}_0$:

$$\mathbf{R}_x \mathbf{W}_0 = \gamma \mathbf{W}_0. \tag{64}$$

Thus, $\mathbf{W}_0$ must be a unit-norm eigenvector of $\mathbf{R}_x$ with eigenvalue γ. Substituting $\mathbf{W} = \mathbf{W_0}$, equations 64, and 61 into 59, we obtain $P_x = \gamma$. Since $\mathbf{R}_x$ is Hermitian and nonnegative definite, $\gamma \geq 0$ and $P_x \geq 0$. Therefore, P_x is minimized when $\gamma = \lambda_{min}$, the smallest eigenvalue $\mathbf{R}_x$. We conclude that the optimal weight vector is a unit-norm eigenvector of $\mathbf{R}_x$ with eigenvalue λ_{min}.

The method of steepest descent for discrete-time systems and equation 63 give the recursive equation for the unconstrained weight vector:

$$\begin{aligned} \mathbf{W}'(k+1) &= \mathbf{W}'(k) - \mu_0(k) \nabla_w H \\ &= [1 + 2\gamma\mu_0(k)]\mathbf{W}'(k) - 2\mu_0(k)\mathbf{R}_x \mathbf{W}'(k) \end{aligned} \tag{65}$$

where $\mu_0(k)$ is a real sequence that regulates the convergence. Defining $\mu_1(k) = 2\mu_0(k)/[1 + 2\gamma\mu_0(k)]$, the recursive equation for the constrained weight vector $\mathbf{W}(k) = \mathbf{W}'(k)/ \parallel \mathbf{W}'(k) \parallel$ is

$$\mathbf{W}(k+1) = \frac{\mathbf{W}(k) - \mu_1(k)\mathbf{R}_x \mathbf{W}(k)}{\parallel \mathbf{W}(k) - \mu_1(k)\mathbf{R}_x \mathbf{W}(k) \parallel}. \tag{66}$$

To convert this equation into a practical algorithm, we observe that if $\mathbf{W}(k)$ is modeled as deterministic, then $\mathbf{R}_x \mathbf{W}(k) = E[\mathbf{x}(i)\mathbf{x}^H(i)]\mathbf{W}(k) = E[\mathbf{x}(i)y^*(i)]$, where i is the sample index, k is the weight iteration index, and the asterisk denotes the complex conjugate. Thus, we only require an estimator for $E[\mathbf{x}(i)y^*(i)]$.

A further simplification that ultimately reduces the amount of computation by nearly a factor of two is obtained by assuming that $\mathbf{x}(i)$ is obtained by sampling a continuous-time vector $\mathbf{X}(t)$ of complex envelopes. Each component of $\mathbf{X}(t)$ is a delayed version of the first component, which is modeled as a zero-mean, wide-sense stationary process. As shown in section 2, this model implies equation 16 and that

$$\mathbf{R}_x \mathbf{W} = 2E[\mathbf{x}(i)y_r(i)]. \tag{67}$$

Substituting this equation into equation 66 and defining $\alpha(k) = 2\mu_1(k)$, we obtain the *recursive suppression algorithm*:

$$\mathbf{W}(k+1) = \frac{\mathbf{W}(k) - \alpha(k)\mathbf{C}_x(k)}{\parallel \mathbf{W}(k) - \alpha(k)\mathbf{C}_x(k) \parallel}, \quad k \geq 0 \tag{68}$$

where $\mathbf{C}_x(k)$ is an estimate of $E[\mathbf{x}(i)y_r(i)]$ and $\mathbf{W}(0)$ is the specified unit-norm initial weight vector.

The vector applied to the adaptive filter is $\mathbf{x}(i)$, and m chip-rate samples are taken per weight iteration. The adaptive filter produces the output

$$y_r(i) = Re[\mathbf{W}^H(k)\mathbf{x}(i)], \quad i = km+1, \cdots, (k+1)m \tag{69}$$

where sample i is taken after weight iteration k. This output is used in the estimator

$$\mathbf{C}_x(k) = \frac{1}{m} \sum_{i=km+1}^{(k+1)m} \mathbf{x}(i) y_r(i), \quad k \geq 0. \tag{70}$$

The adaptation sequence $\alpha(k)$ is chosen so that $\alpha(k)\mathbf{C}_x(k)$ has dimensionless components and so that the mean weight vector converges. As shown subsequently, a suitable choice is

$$\alpha(k) = \frac{\alpha}{\hat{t}_x(k)} \tag{71}$$

where α is the *adaptation constant,* and an estimator of the total power entering the adaptive filter is

$$\hat{t}_x(k) = \frac{1}{m} \sum_{i=km+1}^{(k+1)m} \parallel \mathbf{x}(i) \parallel^2, \quad k \geq 0 \ . \tag{72}$$

Simulation experiments confirm that for cyclostationary spread-spectrum signals and tone interference, the recursive suppression algorithm suffers no degradation from the simplification stemming from the assumption that $X_1(t)$ is wide-sense stationary. As has been verified by simulation, recursive versions of equations 70 and 72 merely slow the convergence of the algorithm, even if the signal statistics are stationary.

6.2 Convergence of Mean Weight Vector

To analyze the convergence of the mean weight vector of the recursive suppression algorithm, the estimate $\hat{t}_x(k)$ is approximated by its mean value $E[\hat{t}_x(k)] = tr(\mathbf{R}_x)$. The norm of $\mathbf{V}(k) = \mathbf{W}(k) - \alpha(k)\mathbf{C}_x(k)$ in equation 68 is approximated by $\parallel E[\mathbf{V}(k)] \parallel$. Both of these approximations become more accurate as k increases. Since $\mathbf{W}(k)$ does not depend on $\mathbf{x}(i)$ for $i \geq km+1$, we make the approximation that $\mathbf{W}(k)$ is statistically independent of $\mathbf{x}(i)$ for $i \geq km+1$. Therefore, equation 16 implies that

$$E[\mathbf{C}_x(k)] = E[\mathbf{x}(i)y_r(i)] \approx \frac{1}{2}E[\mathbf{x}(i)\mathbf{x}^H(i)]E[\mathbf{W}(k)] = \frac{1}{2}\mathbf{R}_x E[\mathbf{W}(k)]. \tag{73}$$

Taking the expected values of both sides of equation 68 and using the approximations, we obtain

$$E[\mathbf{W}(k+1)] = \frac{(\mathbf{I} - \zeta\mathbf{R}_x)E[\mathbf{W}(k)]}{\parallel (\mathbf{I} - \zeta\mathbf{R}_x)E[\mathbf{W}(k)] \parallel}\ , \qquad k \geq 0 \tag{74}$$

where

$$\zeta = \frac{\alpha}{2tr(\mathbf{R}_x)}. \tag{75}$$

By mathematical induction, equation 74 implies that

$$E[\mathbf{W}(k)] = \frac{(\mathbf{I} - \zeta \mathbf{R}_x)^k \mathbf{W}(0)}{\parallel (\mathbf{I} - \zeta \mathbf{R}_x)^k \mathbf{W}(0) \parallel} \ , \qquad k \geq 0. \tag{76}$$

Since $\mathbf{R}_x$ is Hermitian, it has a complete set of orthonormal eigenvectors. For an $N \times N$ matrix $\mathbf{R}_x$, let $S_{\min}$ denote the space spanned by the $N - M$ orthonormal eigenvectors associated with λ_{min}, the smallest eigenvalue, and let S_M denote the space spanned by the M remaining orthonormal eigenvectors. An arbitrary initial weight vector can be decomposed as

$$\mathbf{W}(0) = \mathbf{u} + \sum_{i=1}^{M} \beta_i \mathbf{v}_i \tag{77}$$

where $\mathbf{u}$ is in $S_{\min}$, the $\{\beta_i\}$ are scalars, and $\mathbf{v}_i$, $i = 1, 2, \cdots, M$, is the orthonormal set of eigenvectors in S_M. The probability that an arbitrarily chosen nonzero $\mathbf{W}(0)$ will have $\mathbf{u} = \mathbf{0}$ is zero, so we ignore this possibility. Since

$$\mathbf{u}^H \mathbf{v}_i = \mathbf{v}_i^H \mathbf{v}_j = 0, \qquad i, j = 1, 2, \cdots, M, \quad i \neq j \tag{78}$$

the substitution of equations 77 into 76 yields

$$E[\mathbf{W}(k+1) = \frac{(1 - \zeta \lambda_{min})^k \mathbf{u} + \sum_{i=1}^{M} \beta_i (1 - \zeta \lambda_i)^k \mathbf{v}_i}{\left[(1 - \zeta \lambda_{min})^{2k} \parallel \mathbf{u} \parallel^2 + \sum_{i=1}^{M} \beta_i^2 (1 - \zeta \lambda_i)^{2k}\right]^{1/2}} \tag{79}$$

where λ_i is the eigenvalue associated with $\mathbf{v}_i$.

If every coefficient of $\mathbf{v}_i$ is eventually exceeded in magnitude by the coefficient of $\mathbf{u}$ as k increases, then $E[\mathbf{W}(k+1)]$ converges to an optimal value. Thus, if

$$|1 - \zeta \lambda_i| < |1 - \zeta \lambda_{\min}|, \qquad i = 1, 2, \cdots, M \tag{80}$$

then

$$\lim_{k \to \infty} E[\mathbf{W}(k)] = \frac{\mathbf{u}}{\parallel \mathbf{u} \parallel} \ . \tag{81}$$

Conversely, if equation 81 holds for arbitrary $\mathbf{W}(0)$ and, hence, an arbitrary choice of the β_i, $i = 1, 2, \cdots, M$, then equation 80 must be satisfied. Thus, equation 80 is a necessary and sufficient condition for the convergence of the mean weight vector to an optimal value.

Since $\mathbf{R}_x$ is nonnegative definite, λ_{min} and each λ_i are nonnegative. By separately evaluating equation 80 for the cases in which $\zeta \lambda_i \leq 1$ and $\zeta \lambda_i > 1$, we verify that a sufficient convergence condition is

$$0 < \zeta < \frac{2}{\lambda_{max} + \lambda_{min}} \tag{82}$$

where λ_{max} is the largest eigenvalue of $\mathbf{R}_x$. Since $\lambda_{max} + \lambda_{min} \le tr(\mathbf{R}_x)$, it follows from equations 82 and 75 that

$$0 < \alpha < 4 \tag{83}$$

is a sufficient condition for the convergence of the mean weight vector to an optimal value. The fact that the bounds are numerical and do not depend on environmental parameters provides support for the choice of equation 71 as the adaptation sequence. Simulation experiments indicate that the convergence rate tends to increase as α increases within the bounds of equation 83.

6.3 Simulation of Recursive Suppression Algorithm

Simulation experiments were conducted to discover the extent to which interference can be cancelled while avoiding the cancellation of the desired signal. In most of the simulation experiments, the array consists of four omnidirectional antennas located at the vertices of a square. The edge length is equal to the wavelength λ corresponding to the center frequency of the desired signal, which is 3 GHz. The direct-sequence signal with PSK modulation and a rectangular chip waveform arrives from a direction 20 degrees clockwise from the perpendicular to one of the edges. All signals are assumed to arrive as plane waves with no fading. The direct-sequence signal has a frequency offset equal to 1 kHz after downconversion, which models imperfect frequency synchronization. The signal samples are multiplied by the local spreading sequence, which is not synchronized with the spreading sequence of the received direct-sequence signal. However, the minor effects of the imperfect sampling times and the self-interference *(8)* are not modeled in the simulation for simplicity. The multiplication and the presence of data modulation produce a data-sequence component of the branch output sequence that is independent of the spreading sequence imposed on the interference and noise by the multiplication. Both sequences are randomly generated for each simulation trial at the chip rate of 10 Mbps. As indicated in figure 2, the sampling is done once per spreading-sequence chip. The thermal noise in each branch output is modeled as bandlimited white Gaussian noise. The SNR is assumed to be only 0 dB because of the absence of successful despreading of the desired signal and data-bandwidth filtering prior to code acquisition, which also makes the processing gain irrelevant. Weight iterations occur once per 100 spreading-sequence chips. Each simulation trial represents the execution of the algorithm until it terminates. For each trial, the initial weight vector of the adaptive processor is $\mathbf{W}(0) = [1 \;\; 0 \;\; 0 \;\; 0]$, which forms an omnidirectional array pattern. The principal system parameter values are listed in table 4.

The adaptive-array output is assumed to be applied to a serial-search acquisition system designed to achieve code acquisition within 400 μs with a very high probability if its input SINR is maintained above -6 dB (cf. *(8)*, chapter 4). Thus, the number of weight

Table 4. Basic system parameters.

Parameter	**Value**
Array antennas	4 at vertices of square
Center frequency	3 GHz
Array edge length	wavelength
Direction of desired signal	20°
SNR in each branch output	0 dB
Modulation	PSK
Frequency offset of desired signal	1 kHz
Sampling and chip rate	10 Msamples/s
Chips per weight iteration	100
Weight iterations per trial	100
Adaptation constant	$\alpha = 0.08$
Initial weight vector	$[1 \quad 0 \quad 0 \quad 0]$
Frequency offset of interference	10 kHz

iterations per trial is 100, which corresponds to an algorithm duration $T_0 = 1$ ms. This duration allows 600 μs for enough interference suppression to attain an SINR above -6 dB and another 400 μs for acquisition to be completed. The choice of the adaptation constant as $\alpha = 0.08$ is found to produce an algorithm convergence rate that provides rapid interference cancellation with little desired-signal cancellation when the initial SINR is sufficiently low. Simulation experiments indicate that the adaptation constant should be chosen to be inversely proportional to the algorithm duration. Thus, if T_0 were equal to 4 ms, then $\alpha = 0.02$ would be a good choice.

Each of 1, 2, or 3 interference signals is a tone (continuous-wave signal), which generally degrades a direct-sequence system more than noise interference with the same power *(8)*. After the downconversions, the tones have different initial phase shifts and residual frequency offsets equal to 10 kHz, which reflects the mismatch of the tone frequencies and the carrier frequency of the direct-sequence signal. The SINR at the processor output is calculated after each sample time and then averaged over all samples in the time interval between a weight iteration and a preceding weight iteration to determine the SINR at each weight iteration.

Figure 9 illustrates the SINR variation versus the weight iteration number for a typical simulation trial in which one interference tone arrives at a 60° angle with an ISR of 10 dB. For this trial, the SINR is above -6 dB for more than 800 μs, making acquisition almost assured. The array gain pattern after weight iteration k is defined by equation 57. Figure 10 depicts the gain pattern at the end of the simulation trial of figure 9. A null deeper than -20 dB in the direction of the interference source has formed along with a grating null and a partial grating null.

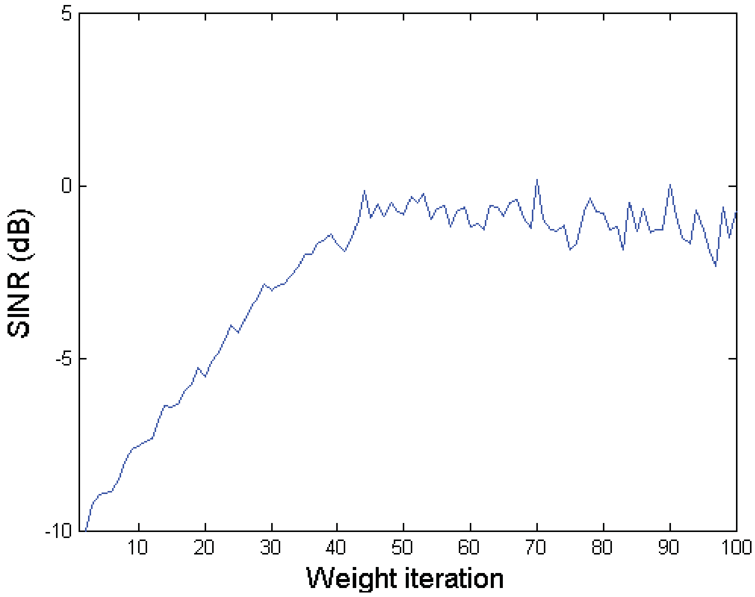

Figure 9. SINR variation in typical simulation trial for recursive suppression algorithm, one interference tone at 60°, and ISR = 10 dB.

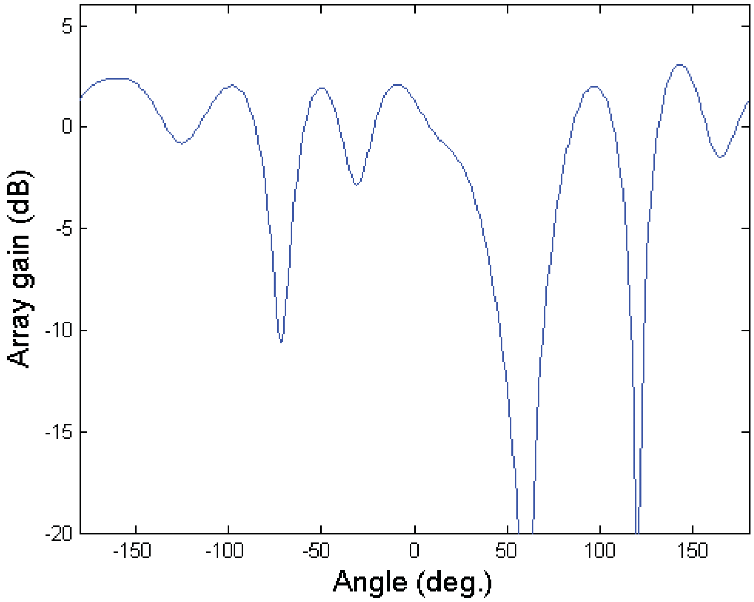

Figure 10. SINR variation in typical simulation trial for recursive suppression algorithm, one interference tone at 60 °, and ISR = 10 dB.

Table 5. Simulation results for interference tones, each with ISR = 10 dB, and square array with $d = \lambda$.

Arrival angles of Interference (°)	**Final SINR (dB)**	**Standard dev. (dB)**	**Relative gain (dB)**
60	−0.68	0.47	9.73
60, −40	−3.22	1.70	16.82
60, 85	−3.26	1.23	16.78
60, −40, 85	−3.76	2.22	26.24
60, −40, 105	−1.46	1.22	29.54
40	−5.39	0.56	5.02

The results of 6 representative simulation experiments using the parameter values of table 4 are summarized in table 5. Each experiment comprises 50 trials with 100 weight iterations per trial. The first column gives the arrival angles of 1, 2, or 3 interference signals relative to the desired-signal direction. The ISR for each interference signal is 10 dB. The SINRs expressed in decibels for the last 40 weight iterations of all the trials are used to compute the *final SINR* and the *standard deviation* of the SINR, which are listed in the second and third columns. The fourth column lists the gain of the final SINR compared with the initial SINR.

The first row of the table shows the system response to a single interference signal separated by 40° from the desired signal. As illustrated in figures 9 and 10, the interference signal is largely suppressed while the desired signal endures much less cancellation. The second and third rows indicate that there is some decrease in the final SINR due to the presence of a second equal-power interference signal, but both interference signals are substantially cancelled. The fourth and fifth rows illustrate the surprising result that the final SINR and the standard deviation (SD) do not necessarily degrade when a third equal-power interference signal is present because both grating lobes and grating nulls are naturally generated as interference signals are nulled. The large values of SD in the second, third, and fourth rows indicate that for some trials the SINR is not maintained above −6 dB for 400 μs, as desired. Therefore, algorithm repetitions may be needed for code acquisition to be achieved.

The final row illustrates the limitations imposed by the *resolution of the array*, which is the minimum angular separation between the interference and desired-signal sources that can be accommodated without significant performance degradation. If the array edge length or antenna separation is increased to $d = 1.5\lambda$, the resolution improves, and the final SINR increases by 3.78 dB relative to the SINR for $d = \lambda$. However, for the scenario of the first row, the increased antenna separation causes the formation of a partial grating null near the direct-sequence signal that reduces the final SINR by 3.51 dB.

If interference arrives at angles in the interval $(-90^\circ, +90^\circ)$, then the resolution can be improved without deep partial gating nulls in the interval by using a uniform linear array of 4 antennas with $d = 0.5\lambda$. The improved resolution is due to the aperture of the linear array, which is 1.5λ. Table 6 lists the results for simulation examples similar to those in table 5 except for the array geometry.

Table 6. Simulation results for interference tones, each with ISR = 10 dB, and uniform linear array with $d = 0.5\lambda$.

Arrival angles of Interference (°)	**Final SINR (dB)**	**Standard dev. (dB)**	**Relative gain (dB)**
60	−1.17	0.47	9.24
60, −40	−0.44	1.64	19.60
60, 85	−1.05	1.38	19.00
60, −40, 85	−0.77	1.35	29.23
40	−0.01	0.48	10.40

Table 7 lists the simulation results for the parameter values of table 4, one interference tone with a 60° arrival angle, and a variable ISR. Each experiment comprises 50 trials with 100 weight iterations per trial. As the ISR increases from 3 dB to 30 dB, the convergence of the recursive suppression algorithm slows, but the final SINR and the relative gain increase monotonically while the standard deviation remains small until ISR> 20 dB. Although an SINR loss or an acquisition failure may occur when ISR< 6 dB, the receiver can acquire the direct-sequence signal by processing the output of a single antenna between algorithm repetitions or whenever a low ISR is known to exist.

Table 7. Simulation results for one interference tone at 60° and square array with $d = \lambda$.

ISR (dB)	**Final SINR (dB)**	**Standard dev. (dB)**	**Relative gain (dB)**
3	−4.97	1.05	−0.20
6	−2.67	0.69	4.32
10	−0.68	0.47	9.73
20	−0.02	0.99	20.02
30	−3.84	2.85	26.16

The results in table 7 are for a fixed adaptation constant $\alpha = 0.08$, which reflects the fact that the interference environment is generally unknown and time varying. However, the best choice for the adaptation constant depends on the ISR, and $\alpha = 0.08$ is close to optimal only for ISR $\approx$ 10 dB. The low final SINR and high SD for ISR = 30 dB are due to the insufficiently rapid cancellation of this strong an interference signal. If the adaptation constant is chosen to be $\alpha = 0.16$ when ISR= 30 dB, then the final SINR = 1.31 dB, SD = 0.44 dB, and the relative gain is 31.31 dB. At the other extreme, if ISR = 3 dB and $\alpha = 0.04$, then the final SINR = -3.74 dB, SD = 0.47 dB, and the relative gain is 1.03 dB. These results indicate the benefit of slowing the algorithm convergence when the interference is not much stronger than the desired signal.

7. Conclusions

An adaptive array using the maximin algorithm provides a direct-sequence system with a high degree of protection against strong interference that would overwhelm the inherent processing gain. The main advantages of the maximin algorithm, relative to its alternatives, are that it does not require training sequences, decision-directed adaptation, directional information, or elaborate computations.

The substantial interference-suppression capability of a direct-sequence system depends on its ability to first achieve code acquisition. However, in the presence of strong directional interference, perhaps from several sources, the code-acquisition process may be severely impaired or completely thwarted. The recursive suppression algorithm appears to be the only existing adaptive-array algorithm that can often suppress strong interference in a direct-sequence system enough to enable rapid code acquisition. The algorithm can be executed periodically over short time intervals to provide repeated acquisition opportunities. During the interims between algorithm executions, the output of a single antenna can be processed to enable acquisition when the interference is weak or absent.

References

[1] Torrieri, D.; Bakhru, K. The Maximin Algorithm for Adaptive Arrays and Frequency-Hopping Systems. *Proc. SPIE* **2001**, *4395,* 119–136.

[2] Tanaka, T. S.; Harada, A.; Sawahashi, M.; Adachi, F. Experiments on Coherent Adaptive Antenna Array Diversity for Wideband DS-CDMA Mobile Radio. *IEEE J. Select. Areas Commun.* **2000**, *18*, 1495–1504.

[3] Song, Y. S.; Kwon, H. M.; Min, B. J. Computationally Efficient Smart Antennas for CDMA Wireless Communications. *IEEE Trans Veh. Technol.* **2001**, *50*, 1613–1628.

[4] Compton, R. T. *Adaptive Antennas: Concepts and Performance.* Prentice-Hall: New York, 1988.

[5] Manolakis, D. G.; Ingle, V. K.; Kogon, S. M. *Statistical and Adaptive Signal Processing.* McGraw-Hill: New York, 2000.

[6] Haykin, S. *Adaptive Filter Theory, 4th ed.* Prentice-Hall: Upper Saddle River, NJ, 2002.

[7] Proakis, J. G. *Digital Communications, 4th ed.* McGraw-Hill: New York, 2001.

[8] Torrieri, D. *Principles of Spread-Spectrum Communication Systems.* Springer: Boston, MA, 2005.

[9] Bensley, S. E.; Aazhang, B. Subspace-based Channel Estimation for Code Division Multiple Access Communications Systems. *IEEE Trans. Commun.* **1996**, *44*, 1009–1020

[10] Wang, B.; Kwon, H. M. PN Code Acquisition for DS/CDMA Systems Employing Smart Antennas–Part II. *IEEE Trans. Wireless Commun.* **2003**, *2*, 108–117.

INTENTIONALLY LEFT BLANK.

A. Appendix - Optimization of the SINR

The signal-to-interference-plus-noise ratio (SINR) at the adaptive-filter output is

$$\rho_0 = \frac{\mathbf{W}^H \mathbf{R}_s \mathbf{W}}{\mathbf{W}^H \mathbf{R}_n \mathbf{W}} \,. \tag{A-1}$$

Since their definitions ensure that $\mathbf{R}_s$ and $\mathbf{R}_n$ are Hermitian and nonnegative definite, these $N \times N$ matrices have complete sets of orthonormal eigenvectors with nonnegative real-valued eigenvalues. The noise power is assumed to be positive, which implies that $\mathbf{R}_n$ is positive definite and has positive eigenvalues. The spectral theorem of linear algebra indicates that $\mathbf{R}_n$ can be expressed as

$$\mathbf{R}_n = \sum_{i=1}^{N} \lambda_i \mathbf{e}_i \mathbf{e}_i^H \tag{A-2}$$

where λ_i is an eigenvalue and $\mathbf{e}_i$ is the associated eigenvector.

To derive the weight vector that maximizes the SINR with no restriction on $\mathbf{R}_s$, we define the Hermitian matrix

$$\mathbf{A} = \sum_{i=1}^{L} \sqrt{\lambda_i} \mathbf{e}_i \mathbf{e}_i^H. \tag{A-3}$$

Direct calculations verify that

$$\mathbf{R}_n = \mathbf{A}^2 \tag{A-4}$$

and the inverse of $\mathbf{A}$ is

$$\mathbf{A}^{-1} = \sum_{i=1}^{L} \frac{1}{\sqrt{\lambda_i}} \mathbf{e}_i \mathbf{e}_i^H. \tag{A-5}$$

The matrix of $\mathbf{A}$ specifies an invertible transformation of $\mathbf{W}$ into the vector

$$\mathbf{V} = \mathbf{A}\mathbf{W}. \tag{A-6}$$

We define the Hermitian matrix

$$\mathbf{C} = \mathbf{A}^{-1} \mathbf{R}_s \mathbf{A}^{-1}. \tag{A-7}$$

Then equations A-1, A-4, A-6, and A-7 indicate that the SINR can be expressed as a *Rayleigh quotient*:

$$\rho_0 = \frac{\mathbf{V}^H \mathbf{C} \ \mathbf{V}}{\parallel \mathbf{V} \parallel^2}. \tag{A-8}$$

Let $\ell_1, \ell_2, \cdots, \ell_N$ and $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N$ denote the eigenvalues and corresponding orthonormal eigenvectors of $\mathbf{C}$. If $\mathbf{V}$ is expanded as

$$\mathbf{V} = \sum_{i=1}^{N} b_i \mathbf{u}_i \tag{A-9}$$

where the $\{b_i\}$ are coefficients, then

$$\mathbf{V}^H \mathbf{C}\ \mathbf{V} = \sum_{i=1}^{N} |b_i|^2 \ell_i \leq \ell_{max} \sum_{i=1}^{N} |b_i|^2 = \ell_{max} \parallel \mathbf{V} \parallel^2 \tag{A-10}$$

where ℓ_{max} is the largest eigenvalue. Therefore, $\rho_0 \leq \ell_{max}$. Direct substitution indicates that ρ_0 is maximized by $\mathbf{V} = \eta \mathbf{u}$, where $\mathbf{u}$ is an eigenvector of $\mathbf{C}$ associated with eigenvalue l_{max}, and η is an arbitrary constant. Thus, the maximum value of ρ_0 is

$$\rho_{max} = l_{max}\ . \tag{A-11}$$

From equation A-6 with $\mathbf{V} = \eta \mathbf{u}$, it follows that an *optimal weight vector that maximizes the SINR* is

$$\mathbf{W}_0 = \eta \mathbf{A}^{-1} \mathbf{u}\ . \tag{A-12}$$

The purpose of an adaptive-array algorithm is to adjust the weight vector to converge to the optimal value, which is given by equation A-12 when the maximization of the SINR is the performance criterion.

If a desired signal with input power p_{si} is sufficiently narrowband, then equation 38 indicates that $\mathbf{R}_s$=$p_{si}\mathbf{S}_0\mathbf{S}_0^H$, where $\mathbf{S}_0$ is the steering vector. Substitution into equation A-7 yields

$$\mathbf{C} = p_{si} \mathbf{F}\mathbf{F}^H \tag{A-13}$$

where

$$\mathbf{F} = \mathbf{A}^{-1}\mathbf{S}_0\ . \tag{A-14}$$

The factorization explicitly shows that $\mathbf{C}$ is a rank-one matrix, which has only one nonzero eigenvalue. By direct substitution, it is found that the eigenvector associated with the nonzero eigenvalue is

$$\mathbf{u} = \mathbf{F} = \mathbf{A}^{-1}\mathbf{S}_0 \tag{A-15}$$

and the eigenvalue is

$$\ell_{max} = p_{si} \parallel \mathbf{F} \parallel^2\ . \tag{A-16}$$

Substituting equations A-15 into A-12 and then observing that $\mathbf{R}_n^{-1} = \mathbf{A}^{-2}$, we obtain the optimal weight vector:

$$\mathbf{W}_0 = \eta \mathbf{R}_n^{-1}\mathbf{S}_0 \tag{A-17}$$

where η is an arbitrary constant. The maximum value of the SINR, obtained from equations A-11, A-16, and A-14 is

$$\rho_{max} = p_{si}\mathbf{S}_0^H \mathbf{R}_n^{-1}\mathbf{S}_0\ . \tag{A-18}$$

Distribution

ADMNSTR
DEFNS TECHL INFO CTR
ATTN DTIC-OCP (ELECTRONIC COPY)
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218

DARPA
ATTN IXO S WELBY
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

OFC OF THE SECY OF DEFNS
ATTN ODDRE (R&AT)
THE PENTAGON
WASHINGTON DC 20301-3080

US ARMY TRADOC
BATTLE LAB INTEGRATION & TECHL
DIRCTRT
ATTN ATFEDS
10 WHISTLER LANE
FT MONROE VA 23651-5850

SMC/GPA
2420 VELA WAY STE 1866
EL SEGUNDO CA 90245-4659

TECOM
ATTN CSTE-DTC-CL
ABERDEEN PROVING GROUND MD
21005-5057

US ARMY ARDEC
ATTN AMSTA-AR-TD
BLDG 1
PICATINNY ARSENAL NJ 07806-5000

COMMANDING GENERAL
US ARMY AVN & MIS CMND
ATTN AMSAM-RD W C MCCORKLE
REDSTONE ARSENAL AL 35898-5000

US ARMY INFO SYS ENGRG CMND
ATTN AMSEL-IE-TD F JENIA
FT HUACHUCA AZ 85613-5300

US ARMY NATICK RDEC
ACTING TECHL DIR
ATTN SBCN-TP P BRANDLER
KANSAS STREET BLDG 78
NATICK MA 01760-5056

US ARMY SIMULATION TRAIN &
INSTRMNTN CMND
ATTN AMSTI-CG M MACEDONIA
12350 RESEARCH PARKWAY
ORLANDO FL 32826-3726

CUBIC DEFNS SYS
ATTN K BAKHRU
9323 BALBOA AVE
SAN DIEGO CA 92123

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI-OK-TP TECHL
LIB T LANDFRIED (2 COPIES)
ABERDEEN PROVING GROUND MD
21005-5066

DIRECTOR
US ARMY RSRCH LAB
ATTN AMSRD-ARL-RO-D JCI CHANG
ATTN AMSRD-ARL-RO-EN W D BACH
PO BOX 12211
RESEARCH TRIANGLE PARK NC 27709

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI J D GANTT
ATTN AMSRD-ARL-CI J GOWENS
ATTN AMSRD-ARL-CI-OK-T TECHL
PUB (2 COPIES)
ATTN AMSRD-ARL-CI-OK-TL TECHL
LIB (2 COPIES)
ATTN AMSRD-ARL-D J M MILLER
ATTN AMSRD-ARL-CI-C D TORRIERI
(15 COPIES)
ATTN IMNE-ALC-IMS MAIL &
RECORDS MGMT
ADELPHI MD 20783-1197